

Задание 6

Вариант 1

Опишите класс `Fraction`, представляющий собой обыкновенную дробь: класс включает в себя целые элементы-данные `numerator` (числитель) и `denominator` (знаменатель). Реализуйте в классе интерфейс `Comparable`, так, чтобы можно было упорядочивать дроби по возрастанию.

Реализуйте компаратор для упорядочивания дробей так, чтобы можно было бы упорядочивать дроби одним из четырёх способов: по возрастанию, по убыванию, по возрастанию абсолютных значений, по убыванию абсолютных значений.

Продемонстрируйте работу функционала на примере сортировки дробей методом `Collections.sort()`.

Вариант 2

PriorityQueue - это очередь с приоритетом, которая по умолчанию размещает элементы согласно естественному порядку сортировки. Для сортировки здесь используется `Comparable` и `Comparator`.

Смоделируйте с помощью класса-дженерика **PriorityQueue** следующую задачу. Пусть имеется класс *автомобиль*. Для каждого автомобиля определен один из возможных приоритетов – обычный, служебный (полиция), медицинский. Машины подъезжают к заправке (помещаются в очередь с приоритетом). Вначале заправляются (извлекаются из приоритетной очереди) машины с наибольшим приоритетом.

Реализуйте два варианта компаратора для определения приоритета автомобилей. В первом случае высший приоритет - у медицинского автомобиля, во втором – у полиции.

Дополнительное задание. Реализуйте функционал таким образом, чтобы учитывалась также очерёдность поступления автомобилей в очередь. В общем случае при наличии в приоритетной очереди элементов с одинаковым приоритетом они будут извлекаться из неё в произвольном порядке, не связанным с порядком занесения в очередь.